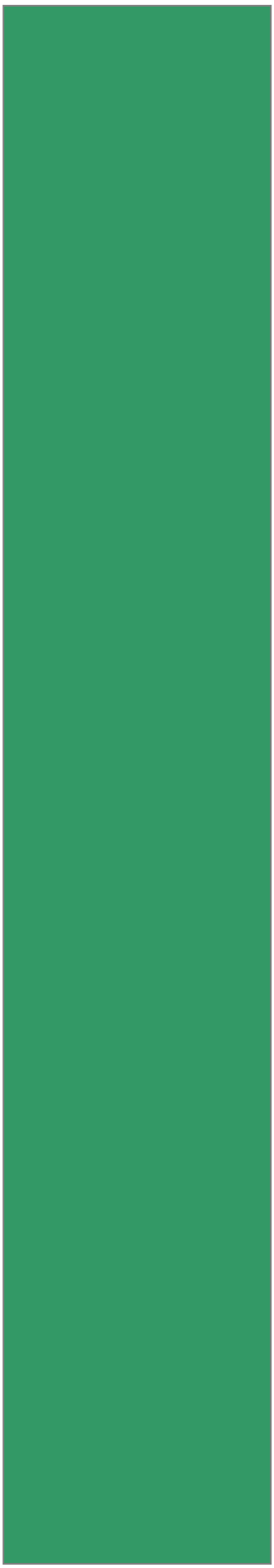


creating value in applied computing

LEC-7900D
User Manual

LEC-7900D
Manual

Version 1.0 : Jane 24, 2011

Copyright 2007, Lanner Electronics, Inc. All rights reserved. This document contains proprietary information that is protected by copyright. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written express of Lanner Electronics, Inc.
The author and Lanner Electronics, Inc. have used their best efforts in preparing this manual. However, the author and Lanner Electronics, Inc. make no warranties of any kind, expressed or implied, with regard to the informational content, documentation, or files contained in this manual, and shall not be liable for technical or editorial errors or omissions contained herein. In no event shall the author or publisher be responsible or liable for any incidental or consequential damages resulting from the furnishing, performance, or use of this material.

TRADEMARKS Internet Explorer, Windows Explorer, and Windows are trademarks or registered trademarks of Microsoft Corporation. Other products mentioned herein may be trademarks/or registered trademarks of their respective owners.

Safety Guidelines

Follow these guidelines to ensure general safety:

- Keep the chassis area clear and dust-free during and after installation.
- Do not wear loose clothing or jewelry that could get caught in the chassis. Fasten your tie or scarf and roll up your sleeves.
- Wear safety glasses if you are working under any conditions that might be hazardous to your eyes.
- Do not perform any action that creates a potential hazard to people or makes the equipment unsafe.
- Disconnect all power by turning off the power and unplugging the power cord before installing or removing a chassis or working near power supplies
- Do not work alone if potentially hazardous conditions exist.
- Never assume that power is disconnected from a circuit; always check the circuit.

Operating Safety

- Electrical equipment generates heat. Ambient air temperature may not be adequate to cool equipment to acceptable operating temperatures without adequate circulation. Be sure that the room in which you choose to operate your system has adequate air circulation.
- Ensure that the chassis cover is secure. The chassis design allows cooling air to circulate effectively. An open chassis permits air leaks, which may interrupt and redirect the flow of cooling air from internal components.

Electrostatic discharge (ESD) can damage equipment and impair electrical circuitry. ESD damage occurs when electronic components are improperly handled and can result in complete or intermittent failures. Be sure to follow ESD-prevention procedures when removing and replacing components to avoid these problems.

- Wear an ESD-preventive wrist strap, ensuring that it makes good skin contact. If no wrist strap is available, ground yourself by touching the metal part of the chassis.
- Periodically check the resistance value of the antistatic strap, which should be between 1 and 10 megohms (Mohms).

LITHIUM BATTERY CAUTION:

- RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE.
- DISPOSE OF USED BATTERIES ACCORDING TO THE INSTRUCTIONS

EMC Notice

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case users will be required to correct the interference at their own expense.

Class A Notice for FCC
Modifying the equipment without the authorization of Lanner Electronics, Inc. may result in the equipment no longer complying with FCC requirements for Class A digital devices. In that event, your right to use the equipment may be limited by FCC regulations, and you may be required to correct any interference to radio or television communications at your own expense.

This equipment is in compliance with the essential requirements and other relevant provisions of Directive 1999/5/EC.

SAFETY GUIDELINES **2**

EMC NOTICE **3**

1 PRODUCT OVERVIEW **5**

1.1 PRODUCT INTRODUCTION 5
1.2 SPECIFICATIONS 6

2 SYSTEM COMPONENTS **8**

2.1 BLOCK DIAGRAM 8
2.2 LEC-7900D(LEB-7900D) SYSTEM BOARD 9
2.3 JUMPER SETTING AND PIN ASSIGNMENT 11

3 HARDWARE INSTALLATION GUIDE **18**

3.1 LEC-7900D EMBEDDED SYSTEM 18
3.1.1 Begin Installation *18*
3.1.2 System Memory Installation *19*
3.1.3 SIM Card Installation *19*
3.1.4 CompactFlash Card Installation *20*
3.1.5 HDD Installation *20*
3.1.6 Heat sink Installation *23*
3.1.7 Mini-PCIexpress Socket Installation *25*
3.1.8 System Complete *25*

APPENDIX A. WATCHDOG TIMER **26**

APPENDIX B. DIGITAL I/O **29**

TERMS AND CONDITIONS **34**

WARRANTY POLICY : 34
RMA SERVICE : 34
Requesting a RMA# *34*
RMA Service Request Form *35*

1 Product Overview

1.1 Product Introduction

Front Panel

Back Panel

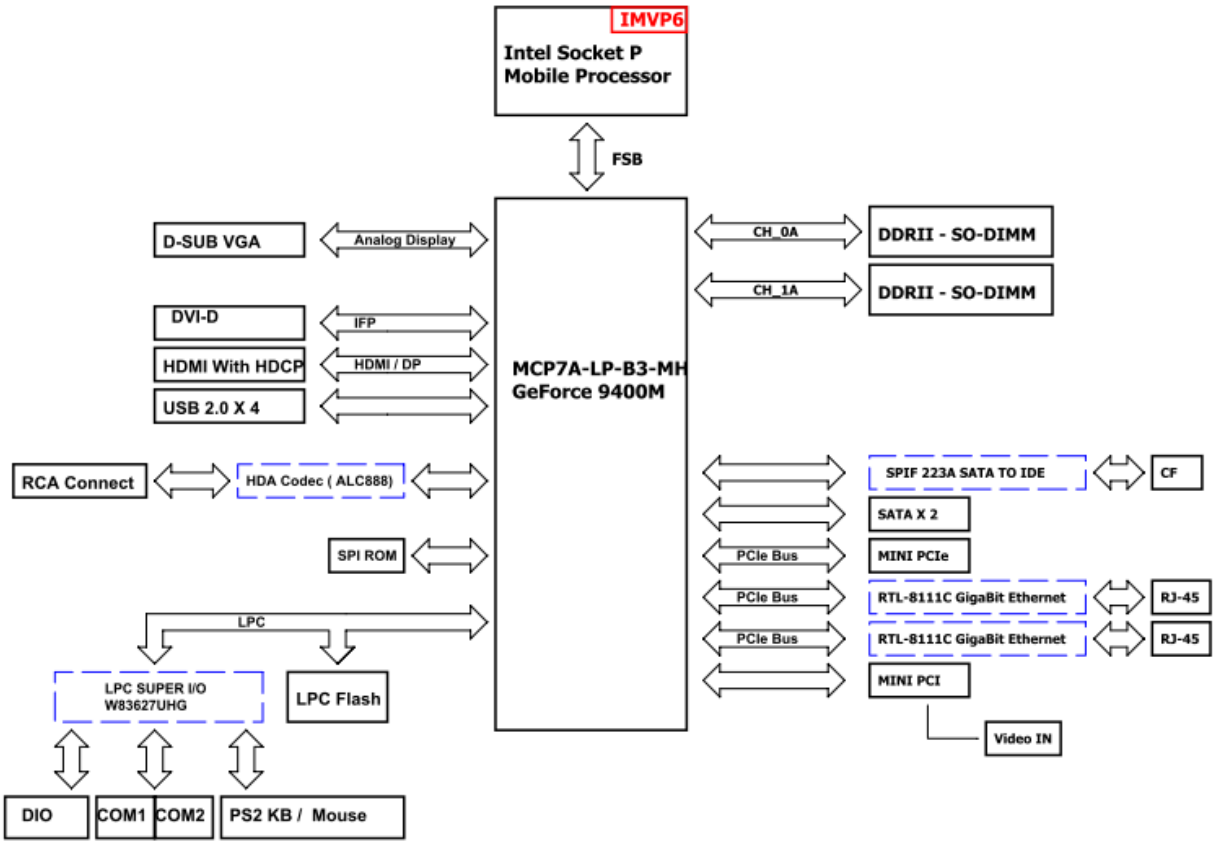
1.2 Specifications

FEATURE	DESCRIPTION	LEC-7900D
Platform	Form Factor	Embedded System
	Processor	Intel Core 2 Duo, Celeron M (Socket P)
	Chipset	Nvidia MCP7A-LP
	BIOS	AMIBIOS with 8Mbit FWH
	Max. FSB	1066MHz
Memory	Memory IC On Board	No
	Memory Socket	SODIMM x 2 (up to 2GB per slot)
	Max Memory	4GB (2GB Module x 2)
Storage	Compact Flash	CF Socket Type I/II x1
	HDD	2.5” SATA HDD support
Networking	Controller (Interface)	RTL8111C(D) (PCIe 1X)
Video	Controller	Nvidia MCP7A-LP
Audio	Codec	ALC888 HD Codec
I/O	COM Ports	RS-232 x 1
	DIO	4 Digital In, 4 Digital Out
	USB 2.0	4 (External x 4)
	Video Grabber / 3G / GPS	Video Grabber (Manufacture Option)
	VGA	2 (1920 x 1440 / 75Hz)
	DVI-D	UXGA (1600 x 1200)
	HDMI	1 x (1.3 Mode 1920 x 1200)
	LAN	RJ45 GbE x 2
	Audio	Stereo Line out (RCA L/R)

	Internal CF	1
	DC-in Connector Type	Phoenix Contact 2-pin, DC Jack (MP Version)
	Expansion	Mini-PCIe x 1
Hardware Monitor	Watchdog timer	Yes (1~255 level)
	Controller	Winbond W83627 UHG integrated hardware monitor
OS Supported		Win XP-32bit/ Windows Vista
Environmental Parameters	Operating Temperature (With Industrial Components: CF, Memory, SSD, HDD)	-10°~55°C / 14°~131°F
	Operating Temperature (With Commercial Components)	-5°C~45°C / 23°~113°F
	Extended Operating Temperature Tested	N/A
Dimensions	W x H x D (mm)	268(W) x 44(H) x 174(D)
	W x H x D (Inch)	10.55”(W) x 1.73”(H) x 6.85”(D)
	Input (Mode)	DC +12V (ATX Mode)
Power	AC Adapter	AC Adapter D: 75W (+12V)
Compliance		CE, FCC, RoHS
Cooling	System Fan	1 x Smart FAN

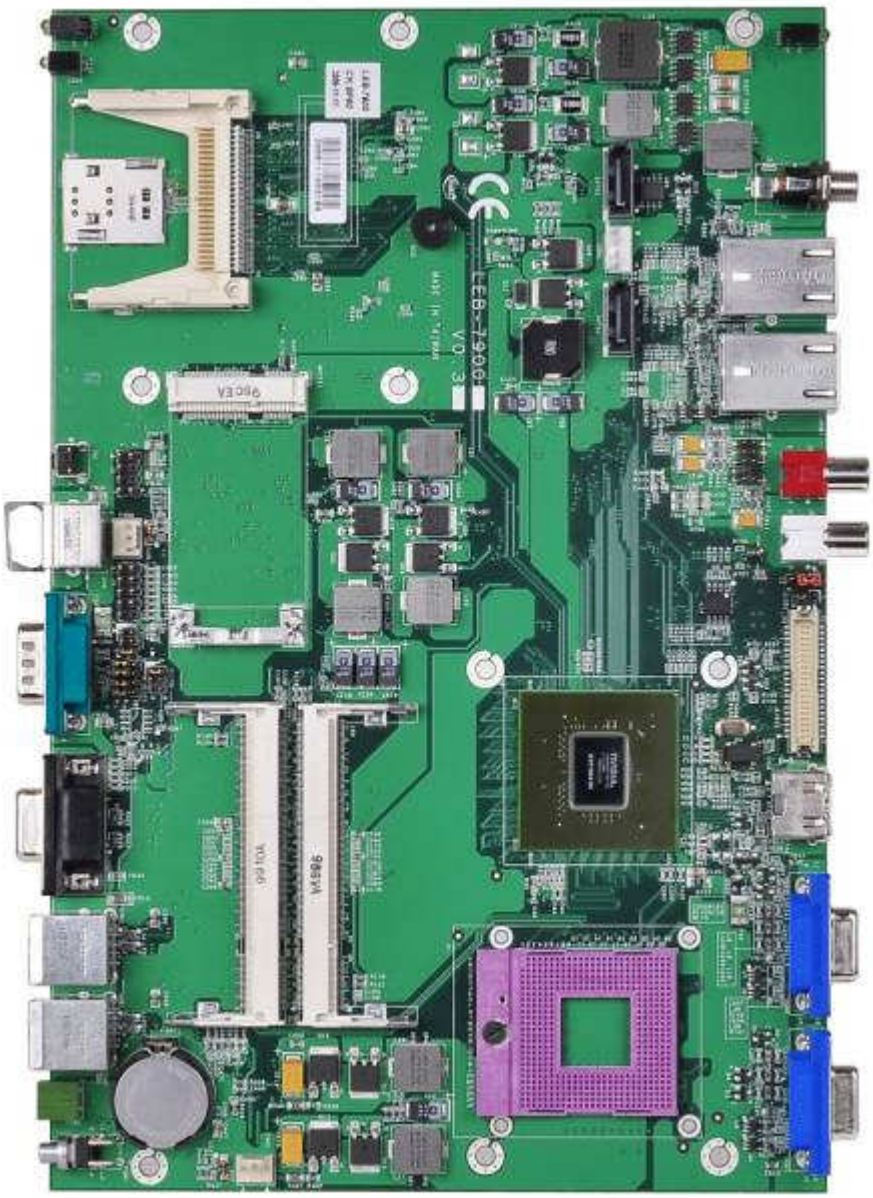
2 System Components

2.1 Block Diagram



2.2 LEC-7900D(LEB-7900D) System Board

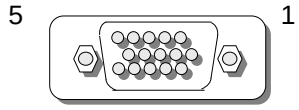
LEB-7900D is the system board bundled with the LEC-7900D Fanless Embedded System platform. The succeeding sections list LEB-7900D related jumper settings and connector pin assignments.



2.3 Jumper setting and Pin Assignment

VGA1: Internal VGA Connector

Pin No.	Description	Pin No.	Description	Pin No.	Description
1	CRT-R	6	GND	11	NC
2	CRT-G	7	GND	12	V_SDAT
3	CRT-B	8	GND	13	HSYNC
4	NC	9	VCC	14	VSYNC
5	GND	10	GND	15	V_SCLK



(DB-15 Female)

CN3: Digital Visual Interface

Pin No.	Description	Pin No.	Description	Pin No.	Description
1	DATA2-	9	DATA1-	17	DATA0-
2	DATA2+	10	DATA1+	18	DATA0+
3	GND	11	GND	19	GND
4	DATA4-	12	DATA3-	20	DATA5-
5	DATA4+	13	DATA3+	21	DATA5+
6	DDC_CLK	14	VCC	22	GND
7	DDC_DAT	15	GND	23	CLK+
8	N.C	16	HP_DET	24	CLK-

CN4: HDMI

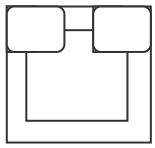
Pin No.	Description	Pin No.	Description	Pin No.	Description
1	TMDS Data 2+	9	TMDS Data 0-	17	GND
2	SHIELD	10	TMDS CLK+	18	5V
3	TMDS Data 2-	11	SHIELD	19	HPD
4	TMDS Data 1+	12	TMDS CLK-		
5	SHIELD	13	CEC		
6	TMDS Data 1-	14	NC		
7	TMDS Data 0+	15	DDC_CLK		
8	SHIELD	16	DDC_DAT		

J5 / J6 : Audio OUT

J5	J6
R-Channel	L-Channel

LANB1 / LANB2

Pin No.	Description	
	Fast E-Net	Giga Net
1	TX+	MD0+
2	TX-	MD0-
3	RX+	MD1+
4	T45	MD2+
5	T45	MD2-
6	RX-	MD1-
7	T78	MD3+
8	T78	MD3-



RJ-45

DC1: DC Power Input

1	2
GND	+12VDC

LED1 : POWER HDD LED

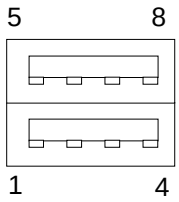
1	2
POWER	HDD

PSW1 : Power Switch

PSW2 : Extern Power Buttom

USB1 / 2 : USB Dual Connector

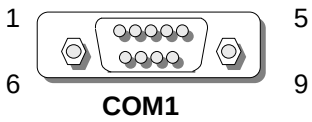
PIN NO.	DESCRIPTION
1	USB_VCC
2	USBD0-
3	USBD0+
4	GND
5	USB_VCC
6	USBD1-
7	USBD1+
8	GND



USBB1

COM1: Internal COM1 (D-SUB9) Connector

PIN NO.	DESCRIPTION
1	Data Carrier Detect (DCDA #)
2	Receive Data (RXDA)
3	Transmit Data (TXDA)
4	Data Terminal Ready (DTRA #)
5	Ground (GND)
6	Data Set Ready (DSRA #)
7	Request To Send (RTSA #)
8	Clear To Send (CTSA #)
9	Ring Indicator (RIA #)

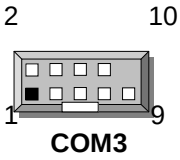


SC2T1 : Select COM1 Type

COM2 TYPE	SC1T2	SC1T1	SC1T3
RS-232 (Default)	1-2	1-5,2-6,3-7,4-8	5-6
RS-422	3-4	5-9,6-10,7-11,8-12	
RS-485	5-6	5-9,6-10,7-11,8-12	

J3: RS-232 Serial Port #2 Connector (2X5 Pin 2.54mm Header)

PIN NO.	DESCRIPTION
	RS-232
1	Data Carrier Detect (DCDB #)
2	Data Set Ready (DSRB #)
3	Receive Data (RXDB)
4	Request To Send (RTSB #)
5	Transmit Data (TXDB)
6	Clear To Send (CTSB #)
7	Data Terminal Ready (DTRB #)
8	Ring Indicator (RIB #)
9	Ground
10	KEY

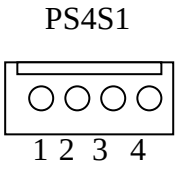


SATA 1 / 2 : SATA CONNECTOR

PIN NO.	DESCRIPTION
1	GND
2	TX+
3	TX-
4	GND
5	RX-
6	RX+
7	GND

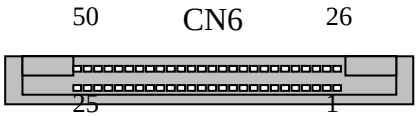
CON1 : 4-Pin Power Connector (Small-4P)

Pin No.	Description
1	5V
2	Ground
3	Ground
4	12V



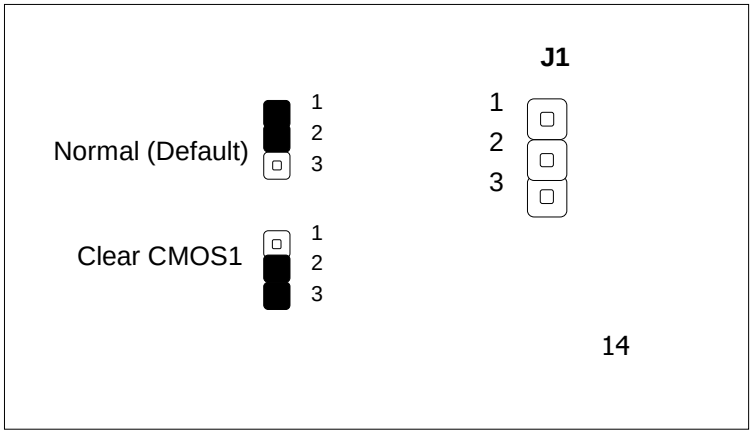
CN6 : Compact Flash Connector

PIN	DESCRIPTION	PIN	DESCRIPTION
1	GND	26	CD1-
2	DATA3	27	DATA11
3	DATA4	28	DATA12
4	DATA5	29	DATA13
5	DATA6	30	DATA14
6	DATA7	31	DATA15
7	CE1#	32	CE2#
8	A10	33	VS1#
9	OE#	34	IOR#
10	A9	35	IOW#
11	A8	36	WE#
12	A7	37	READY#
13	CFVCC3	38	CFVCC3
14	A6	39	CSEL
15	A5	40	VS2#
16	A4	41	RESET
17	A3	42	WAIT#
18	A2	43	INPACK#
19	A1	44	REG#
20	A0	45	DASP#
21	DATA0	46	DIAG#
22	DATA1	47	DATA8
23	DATA2	48	DATA9
24	WP	49	DATA10
25	CD2-	50	GND



J1:Clear CMOS Data

Description	CMOS1
Normal (Default)	1-2
Clear CMOS	2-3



JLCD1: LCD Power

Description	**CMOS1**
3.3V(Default)	1-2
5V	2-3

JLVDS1: LVDS 2x20 1.25mm Connector

Pin No.	**Description**	**Pin No.**	**Description**
1	PVDD	2	12V
3	LCD1D0#	4	LCD1D4#
5	LCD1D0	6	LCD1D4
7	PVDD	8	12V
9	LCD1D1#	10	LCD1D5#
11	LCD1D1	12	LCD1D5
13	GND	14	GND
15	LCD1D2#	16	LCD1D6#
17	LCD1D2	18	LCD1D6
19	GND	20	GND
21	LCD1D3#	22	LCD1D7#
23	LCD1D3	24	LCD1D7
25	LCLK1#	26	LCLK2#
27	LCLK1	28	LCLK2
29	ENBLD1	30	BLCON
31	GND	32	GND
33	ENBLD2	34	BLCON2
35	PVDD2	36	GND
37	PVDD2	38	SPD1
39	GND	40	SPCLK1

MPCI1：Mini-PCI Connector

Pin	Description	Pin	Description	Pin	Description	Pin	Description
1	NC	63	VCC3	2	NC	64	FRAME#
3	NC	65	TP	4	NC	66	TRDY#
5	NC	67	SERR#	6	NC	68	STOP#
7	NC	69	GND	8	NC	70	VCC3
9	NC	71	PERR#	10	NC	72	DEVSEL#
11	NC	73	C_BE#1	12	NC	74	GND
13	NC	75	A_D14	14	NC	76	A_D15
15	NC	77	GND	16	NC	78	A_D13
17	PIRQD#	79	A_D12	18	VCC	80	A_D11
19	VCC3	81	A_D10	20	PIRQC#	82	GND
21	NC	83	GND	22	NC	84	A_D9
23	GND	85	A_D8	24	VCC3	86	C_BE#0
25	CK_33M_	87	A_D7	26	PCI_RST#	88	VCC3
27	GND	89	VCC3	28	VCC3	90	A_D6
29	REQ#1	91	A_D5	30	GNT#1	92	A_D4
31	VCC3	93	NC	32	GND	94	A_D2
33	A_D31	95	A_D3	34	PCI_PME#	96	A_D0
35	A_D29	97	VCC	36	NC	98	NC
37	GND	99	A_D1	38	A_D30	100	NC
39	A_D27	101	GND	40	VCC3	102	GND
41	A_D25	103	NC	42	A_D28	104	M66EN
43	NC	105	NC	44	A_D26	106	NC
45	C_BE#3	107	NC	46	A_D24	108	NC
47	A_D23	109	NC	48	MiniPCIDSEL	110	NC
49	GND	111	NC	50	GND	112	NC
51	A_D21	113	NC	52	A_D22	114	GND
53	A_D19	115	NC	54	A_D20	116	NC
55	GND	117	NC	56	PAR	118	NC
57	A_D17	119	NC	58	A_D18	120	NC
59	C_BE#2	121	NC	60	A_D16	122	NC
61	IRDY#	123	VCC	62	GND	124	VCC3

MPCIE1：Mini-PCIE Connector

Pin #	Signal Name	Pin #	Signal Name
51	Reserved	52	+3.3V
49	Reserved	50	GND
47	Reserved	48	+1.5V
45	Reserved	46	LED_WPAN#
43	Reserved	44	LED_WLAN#
41	Reserved	42	LED_WWAN#
39	Reserved	40	GND
37	Reserved	38	USB_D+
35	GND	36	USB_D-
33	PETp0	34	GND
31	PETn0	32	SMB_DATA
29	GND	30	SMB_CLK
27	GND	28	+1.5V
25	PERp0	26	GND
23	PERn0	24	+3.3Vaux
21	GND	22	PERST#
19	Reserved (UIM_C4)	20	Reserved
17	Reserved (UIM_C8)	18	GND
Mechanical Key			
15	GND	16	UIM_VPP
13	REFCLK+	14	UIM_RESET
11	REFCLK-	12	UIM_CLK
9	GND	10	UIM_DATA
7	CLKREQ#	8	UIM_PWR
5	Reserved	6	1.5V
3	Reserved	4	GND
1	WAKE#	2	3.3V

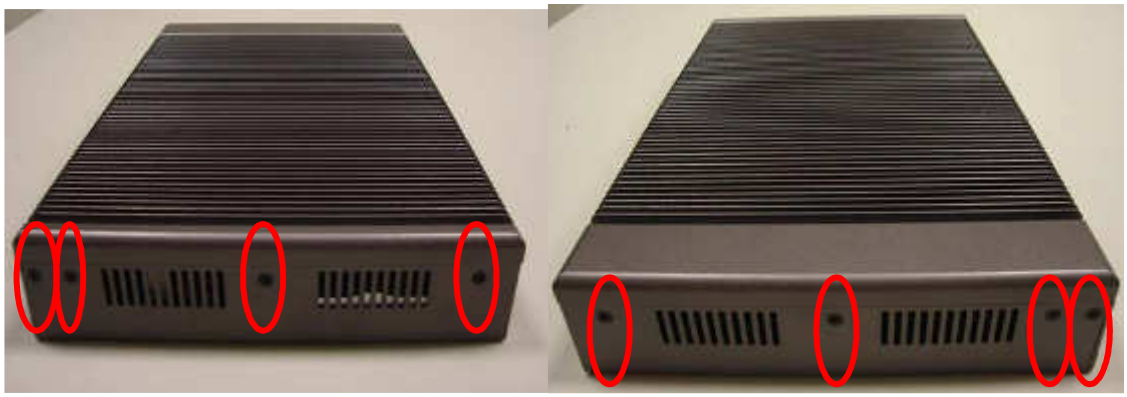
3 Hardware Installation Guide

3.1 LEC-7900D Embedded System

LEC-7900D Embedded System Outlook

3.1.1 Begin Installation

Unscrew 8 thumbscrews of the chassis bottom cover and than remove the chassis bottom cover.



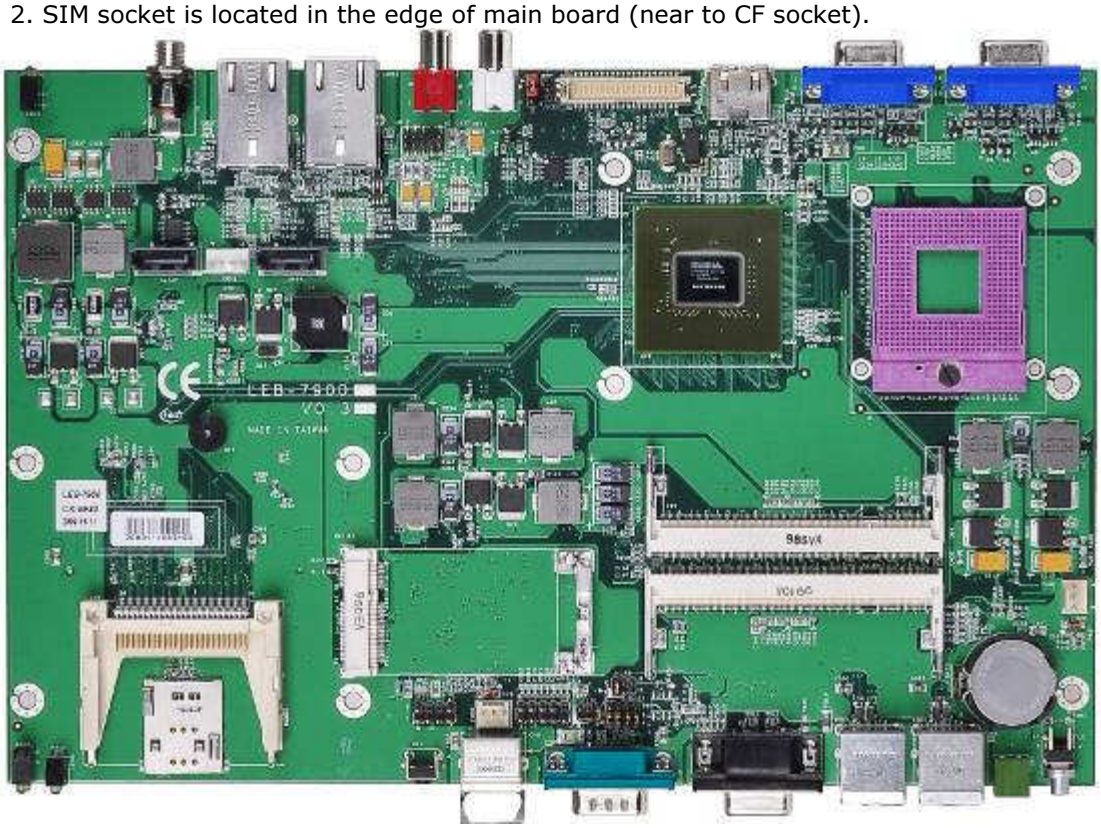
Note: For Safety reasons, please ensure that the power cord is disconnected before opening the case.

3.1.2 System Memory Installation

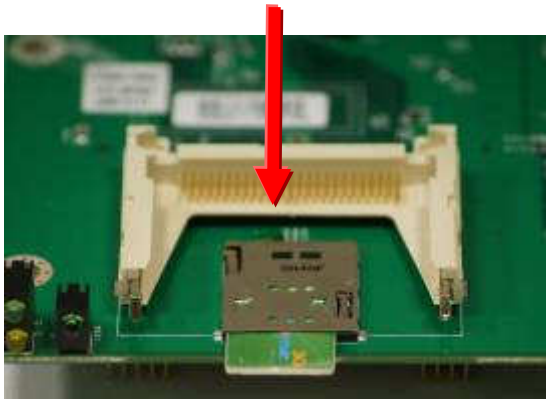
Install the memory module into the socket and push it firmly down until it is fully seated.

3.1.3 SIM Card Installation

1. Remove the Rear panel in order to make room for SIM Card insertion.
2. SIM socket is located in the edge of main board (near to CF socket).



3. SIM card installation

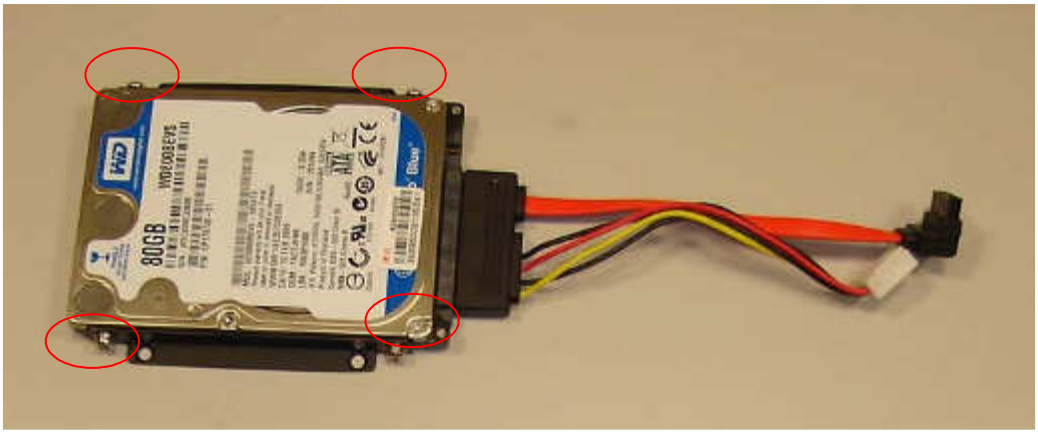


3.1.4 CompactFlash Card Installation

Carefully insert the CompactFlash card into the slot as shown in the illustration above.

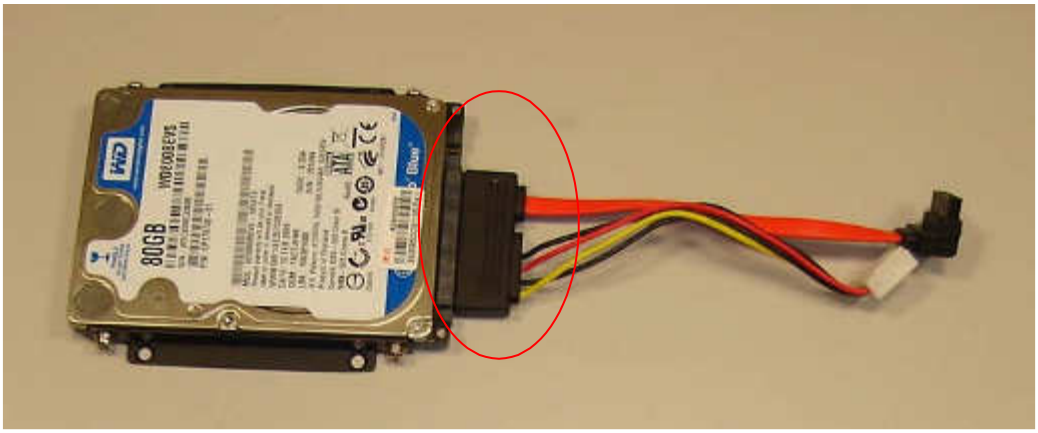
3.1.5 HDD Installation

1. Secure with 4 screws from the side



Note: Only support 2.5” HDD

2. Connect the Serial ATA/Power Cable to the HDD



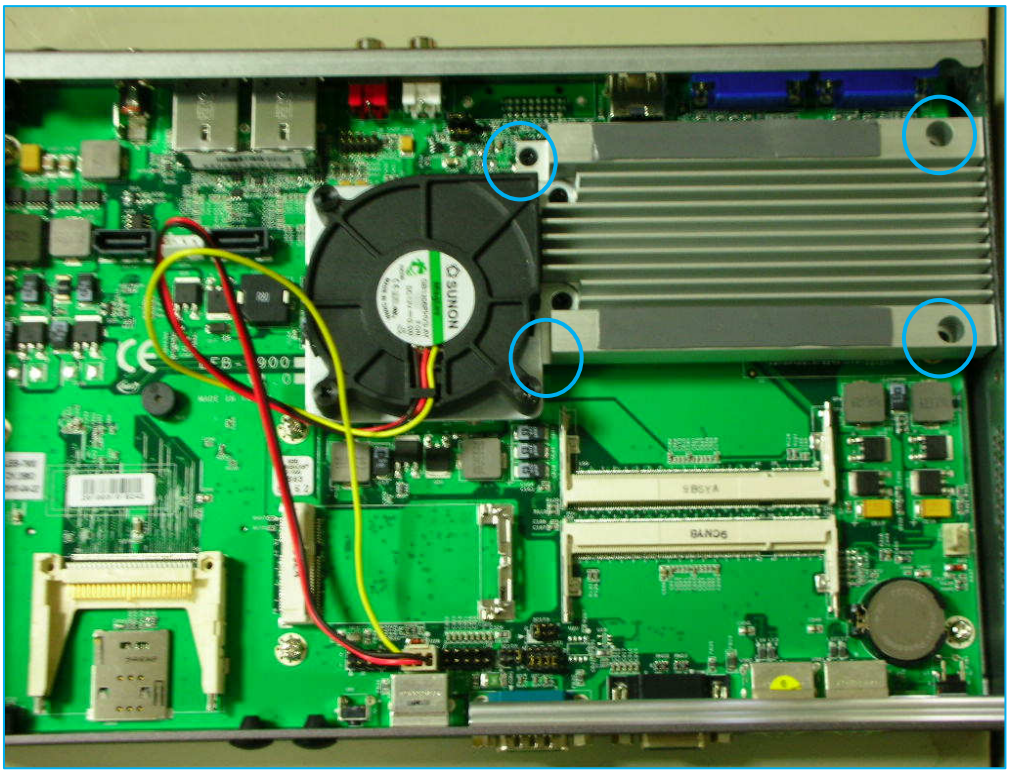
3. Plug the Serial ATA cable to the SATA Connector (SATA 1)

4. Plug the Power cable to the 4-Pin Power Connector (CON1)

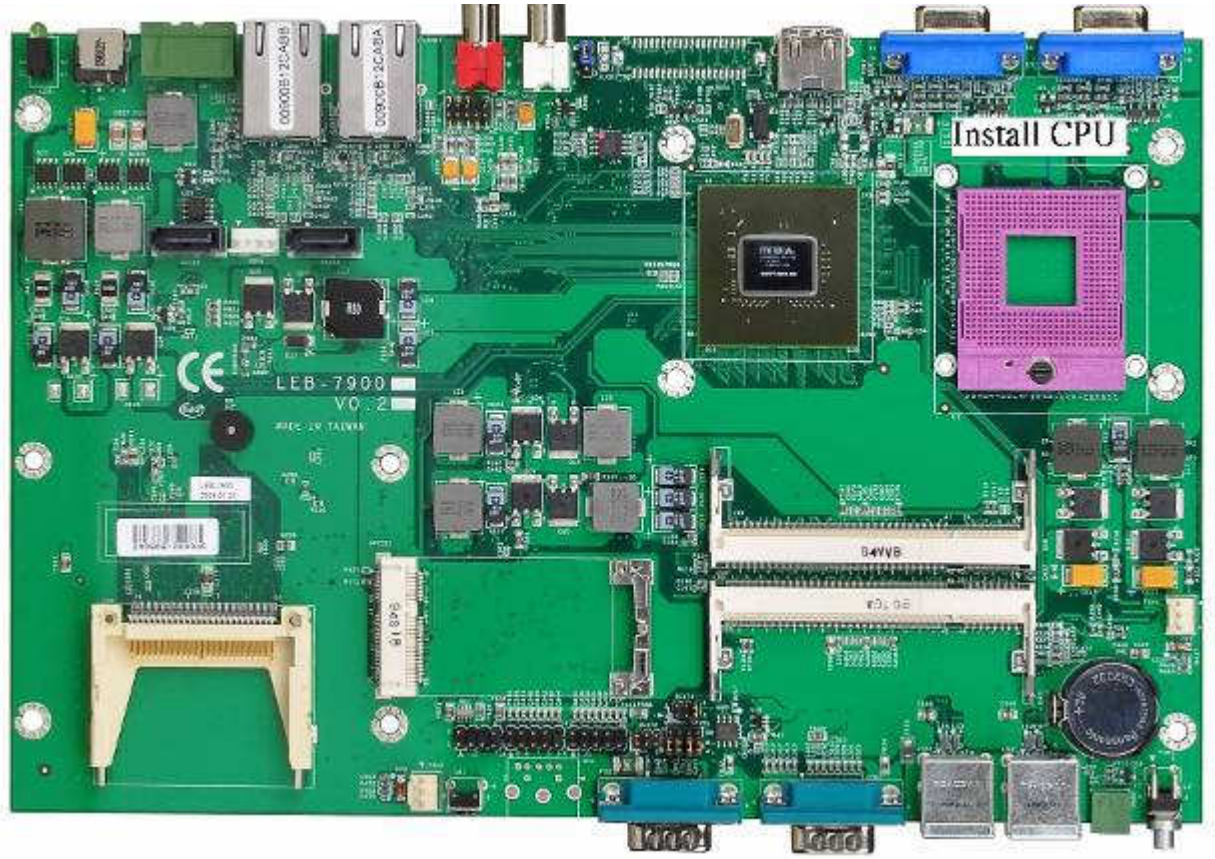
5. Secure with 4 screws from the mother Board.

3.1.6 Heat sink Installation

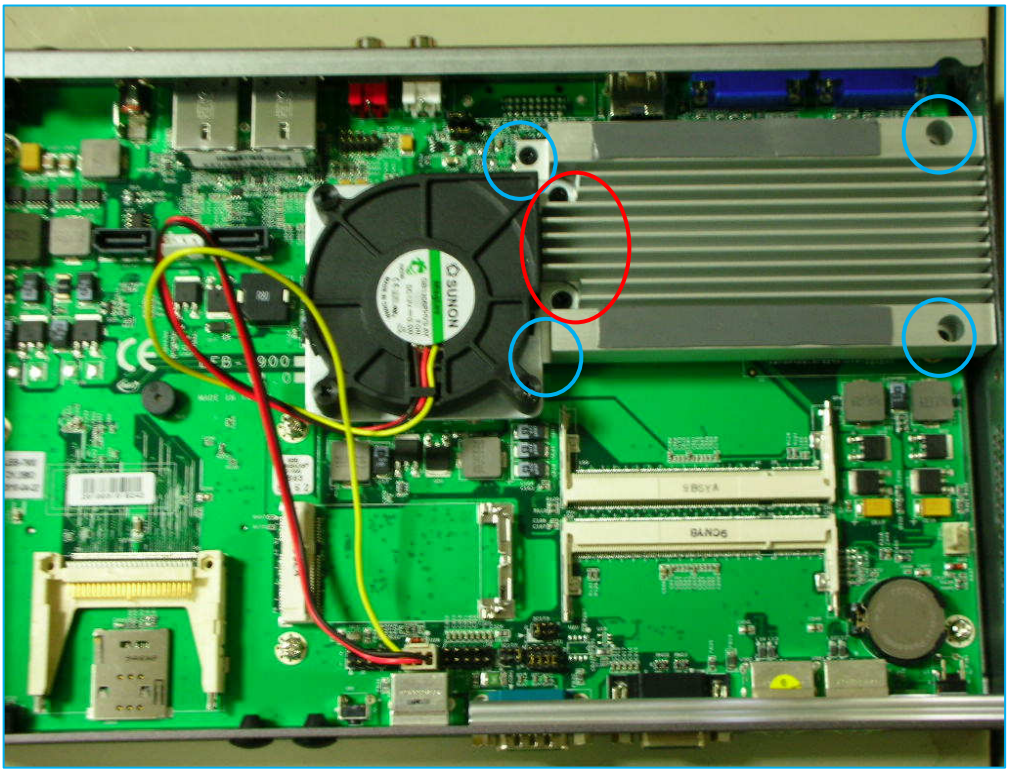
1. Remove LEC-7900D Heat sink.



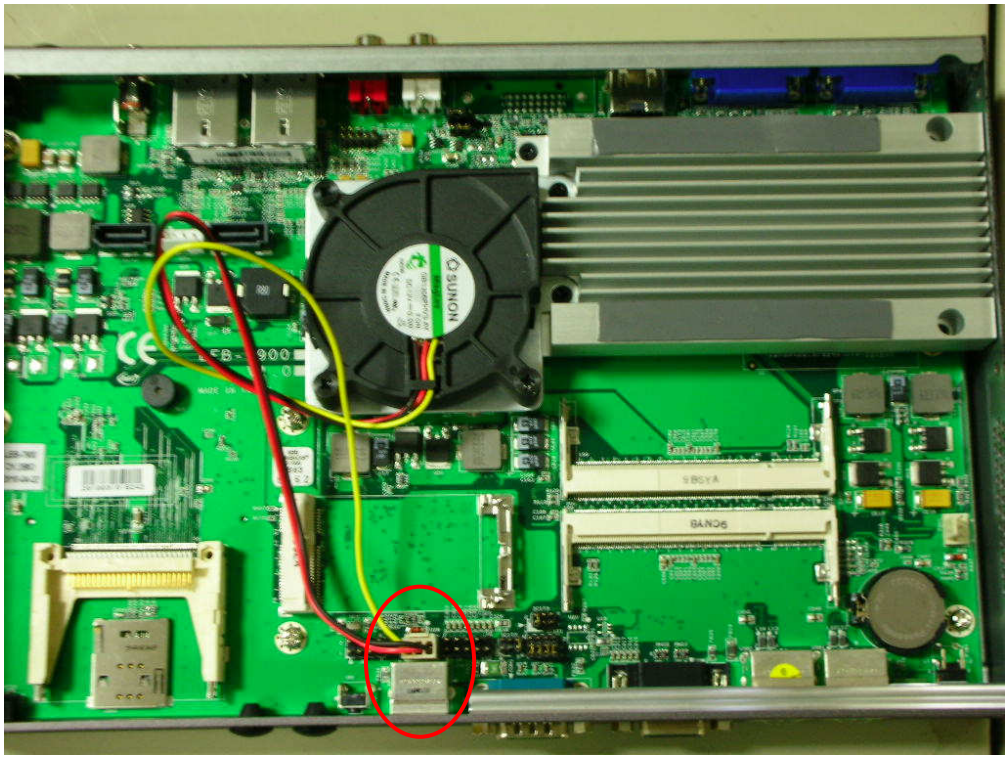
2. Install CPU



3. Secure with 4 screws from the mother Board and 2 screws for the FAN fix.

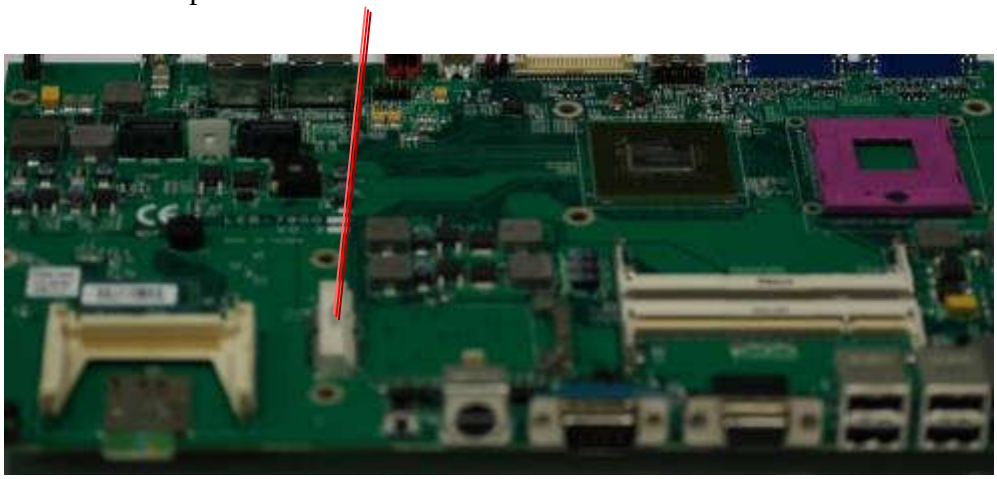


4. Plug the Cooler cable to the FAN Connector (FAN1) as the below red circle



3.1.7 Mini-PCIexpress Socket Installation

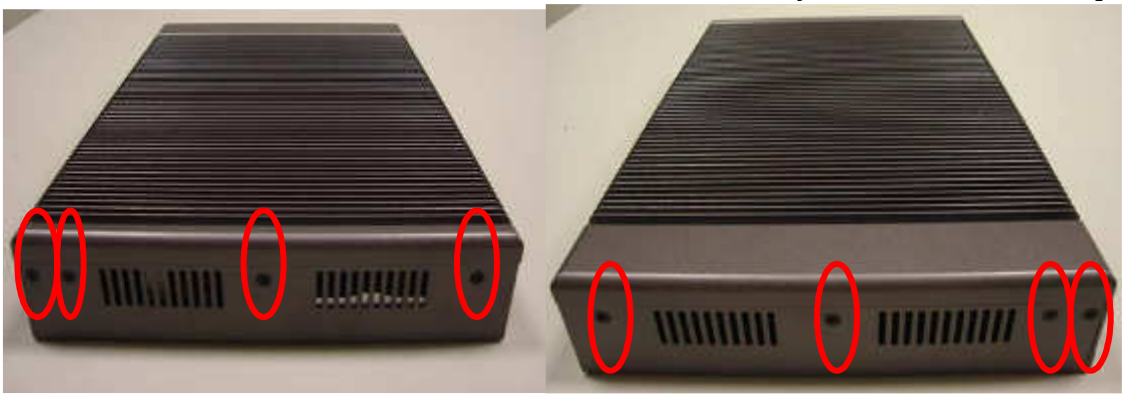
1. Mini-PCIexpress Socket for 3G mini-card module or WiFi mini-card module



3.1.8 System Complete

1. Replace the cover

2. Refasten the thumbscrews and than LEC-7900 Embedded System Installation Complete



Note: Please load the optimized BIOS values.

Appendix A. Watchdog Timer

Introduction

Most systems need to be self-reliant. If an error should occur it is typically not possible to wait for the system to be rebooted manually. In some cases, such as apace probes, the system is simply disabled. In other cases, the speed at which a human operator would reset the system would be too slow to meet the uptime requirements of the product.
A watchdog timer is a piece of hardware that can be used to automatically detect system anomalies and reset the processor if the case any problems are found. Generally speaking, a watchdog timer is based on a counter that counts down from an initial value to zero. The software selects the counter's initial value and periodically restarts it. Should the counter reach zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted. Thus, the processor will be restarted as if a human operator had cycled the power
Watch Dog Sample code:

```
// Copyright (c) 2008 Lanner Electronic Inc. All rights reserved.
// Watchdog control utility for LEB-7900


#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>
#include <inlines/pc.h>

#define INDEX_PORT   0x2e
#define DATA_PORT    0x2f

#defineCR20_REG      0x20    //This register indicate the device's ID
#define W83627UHG_ID         0xA2


void help()
{
    printf("Lanner Electronics Inc.   2009-05-22\n");
    printf("Watchdog timer control V1.0 for LEB-7900\n\n");
    printf("wdbp.exe -wr xxx (1-255 sec)(Watchdog Control - SYSTEM RESET)\n");
    printf("\n");
}

int main(int argc, char *argv[])
{
        int temp9 = 0;
        int time;
        if( argc < 2 )
        {
                help();
                return -1;
        }
        //;********** Detect SIO ******************
        Enter_Configuation();
        if(detect_sio())
        {
                printf("It's not correct SuperIO, program terminated!!!\n");
                temp9 = read_w83627UHG_reg(0, CR20_REG);
                printf("Rx20 = %02X\n", temp9);
                return -1;
        }
        Exit_Configuation();
        Init_WDTO_GPIO_Interface();

//;********************* Setting Watchdog timer for System Reset
```

```
*****************************************************
        if( strcmp(argv[1], "-wr") == 0 )
        {
                if( argc != 3 )
                {
                        help();
                        return -1;
                }
                time = atoi(argv[2]);
                printf("Setting Watchdog timer for System Reset...\n");
                set_watchdog(time);
                return 0;
        }
        else
        {
                help();
                return -1;
        }
        return 0;
}

//;********************* set_watchdog
*****************************************************
int set_watchdog(int time)
{
        int temp;
        temp = time;
        Enter_Configuation();
        write_w83627UHG_reg(0x8, 0x30, 0x01);               //;enable LDN8
        write_w83627UHG_reg(0x8, 0xf6, temp);               //LDN=8, CRF6, set watchdog
timer time-out value
        Exit_Configuation();
        return 0;
}

//;********************* Read SIO
*****************************************************
int read_w83627UHG_reg(int LDN, int reg)
{
        outportb(INDEX_PORT, 0x07); //LDN register
        delay(1);
        outportb(DATA_PORT, LDN);
        delay(1);
        outportb(INDEX_PORT, reg);
        delay(1);
        return inportb(DATA_PORT);
}

//;********************* Write SIO
*****************************************************
int write_w83627UHG_reg(int LDN, int reg, int value)
{
        outportb(INDEX_PORT, 0x07); //LDN register
        delay(1);
        outportb(DATA_PORT, LDN);
        delay(1);
        outportb(INDEX_PORT, reg);
        delay(1);
        outportb(DATA_PORT, value);
        return 0;
}

//;********************* Enter SIO
*****************************************************
int Enter_Configuation()
{
        outportb(INDEX_PORT, 0x87);
        delay(1);
        outportb(INDEX_PORT, 0x87);
```

```
        return 0;
}

//;********************* Exit SIO
*****************************************************
int Exit_Configuation()
{
        outportb(INDEX_PORT, 0xaa);                                                     //Exit
W83627THG Configuration
        return 0;
}

//;********************* Init WDTO and SB GPIO
*****************************************************
int Init_WDTO_GPIO_Interface()
{
        Enter_Configuation();
        write_w83627UHG_reg(0x8, 0x30, 0x01);           //;enable LDN8
        write_w83627UHG_reg(0x8, 0xf5, 0x00);           //;//;set second
        write_w83627UHG_reg(0x8, 0xf6, 0x00);           //;LDN=8, CRF6, stop timer first

        write_w83627UHG_reg(0x8, 0xf7, 0x00);           //;clear status bit
        Exit_Configuation();

        return 0;
}

//input: none
//output: 1 = not correct SuperIO
//          0 = correct SuperIO
int detect_sio(void)
{
        //check the SuperIO version
        if(read_w83627UHG_reg(0,CR20_REG)==W83627UHG_ID)
                return 0;
        else
                return 1;
}
```

Appendix B. Digital I/O

Introduction

The Digitanl I/O on the rear panel is designed to provide the input and output operations for the system For sample DIO code, see Digt-IO folder under LEC-7900D Utility on the Driver and Manual CD.

A DIO sample code:

```
// Copyright (c) 2009 Lanner Electronic Inc. All rights reserved.
// DIO utility for LEB-7900

//;      Output  Input
//;      GPO30  GPI34
//;      GPO31  GPI35
//;      GPO32  GPI36
//;      GPO33  GPI37

#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>
#include <inlines/pc.h>

#define INDEX_PORT   0x2e
#define DATA_PORT    0x2f

#define CR20_REG      0x20    //This register indicate the device's ID
#define W83627UHG_ID        0xA2


void help()
{
    printf("Lanner Electronics Inc.   2009/10/28\n");
    printf("Digital IO control V0.1 for LEB-7900\n");
    printf("\n");
}

int main(int argc, char *argv[])
{
        int temp6, temp8=0;
        help();

        //;********** Detect SIO ******************
        Enter_Configuation();
        if(detect_sio())
        {
                printf("It's not correct SuperIO, program terminated!!!\n");
                printf("Rx20 = %02X\n", read_w83627UHG_reg(0, CR20_REG));
                return -1;
        }
        Exit_Configuation();
        Init_WDTO_GPIO_Interface();

//;********************** DIO start
*****************************************************
        Enter_Configuation();
//;*********************GPO30 and
GPI34*****************************************************
        printf("GPO30(OUT) to GPI34(IN)...\n");
        write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) | 0x01);
        //;Set GPO30 to 1
        delay(50);
        for(temp6=1; temp6<50; temp6++)
```

```
	{
		temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 4) & 0x01;
	//;Read GPI34=?
		delay(50);
		if(temp8 == 1)
		{
			printf("Write OUT GPO30=1   then IN GPI34=%2X \n",temp8);
			goto gpio10;
		}
	}
	printf("GPIO set HI error\n");
gpio10:
	write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) & ~0x01);
	//;Set GPIO30 to 0
	delay(50);
	for(temp6=1; temp6<50; temp6++)
	{
		temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 4) & 0x01;
	//;Read GPIO34=?
		delay(50);
		if(temp8 == 0)
		{
			printf("Write OUT GPO30=0   tnen IN GPI34=%2X \n",temp8);
			goto gpio21;
		}
	}
	printf("GPIO set LO error\n");

//;**********************GPO31 and
GPI35*****************************************************
gpio21:
	printf("GPO31(OUT) to GPI35(IN)...\n");
	write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) | 0x02);
	//;Set GPO31 to 1
	delay(50);
	for(temp6=1; temp6<50; temp6++)
	{
		temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 5) & 0x01;
	//;Read GPI35=?
		delay(50);
		if(temp8 == 1)
		{
			printf("Write OUT GPO31=1   then IN GPI35=%2X \n",temp8);
			goto gpio20;
		}
	}
	printf("GPIO set HI error\n");
gpio20:
	write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) & ~0x02);
	//;Set GPIO31 to 0
	delay(50);
	for(temp6=1; temp6<50; temp6++)
	{
		temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 5) & 0x01;
	//;Read GPIO35=?
		delay(50);
		if(temp8 == 0)
		{
			printf("Write OUT GPO31=0   tnen IN GPI35=%2X \n",temp8);
			goto gpio31;
		}
	}
	printf("GPIO set LO error\n");

//;**********************GPO32 and
GPI36*****************************************************
gpio31:
	printf("GPO32(OUT) to GPI36(IN)...\n");
	write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) | 0x04);
```

```
        //;Set GPO32 to 1
        delay(50);
        for(temp6=1; temp6<50; temp6++)
        {
                temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 6) & 0x01;
        //;Read GPI36=?
                delay(50);
                if(temp8 == 1)
                {
                        printf("Write OUT GPO32=1   then IN GPI36=%2X \n",temp8);
                        goto gpio30;
                }
        }
        printf("GPIO set HI error\n");
gpio30:
        write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) & ~0x04);
        //;Set GPIO32 to 0
        delay(50);
        for(temp6=1; temp6<50; temp6++)
        {
                temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 6) & 0x01;
        //;Read GPIO36=?
                delay(50);
                if(temp8 == 0)
                {
                        printf("Write OUT GPO32=0   tnen IN GPI36=%2X \n",temp8);
                        goto gpio41;
                }
        }
        printf("GPIO set LO error\n");

//;**********************GPO33 and
GPI37*****************************************************
gpio41:
        printf("GPO33(OUT) to GPI37(IN)...\n");
        write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) | 0x08);
        //;Set GPO33 to 1
        delay(50);
        for(temp6=1; temp6<50; temp6++)
        {
                temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 7) & 0x01;
        //;Read GPI37=?
                delay(50);
                if(temp8 == 1)
                {
                        printf("Write OUT GPO33=1   then IN GPI37=%2X \n",temp8);
                        goto gpio40;
                }
        }
        printf("GPIO set HI error\n");
gpio40:
        write_w83627UHG_reg(0x07, 0xE1, read_w83627UHG_reg(0x07, 0xE1) & ~0x08);
        //;Set GPIO33 to 0
        delay(50);
        for(temp6=1; temp6<50; temp6++)
        {
                temp8 = (read_w83627UHG_reg(0x07, 0xE1) >> 7) & 0x01;
        //;Read GPIO37=?
                delay(50);
                if(temp8 == 0)
                {
                        printf("Write OUT GPO33=0   tnen IN GPI37=%2X \n",temp8);
                        goto _end;
                }
        }
        printf("GPIO set LO error\n");
_end:
        printf("Finish!!\n");
        Exit_Configuation();
```

```
        return 0;
}

//;********************* Read SIO
*****************************************************
int read_w83627UHG_reg(int LDN, int reg)
{
        outportb(INDEX_PORT, 0x07); //LDN register
        delay(1);
        outportb(DATA_PORT, LDN);
        delay(1);
        outportb(INDEX_PORT, reg);
        delay(1);
        return inportb(DATA_PORT);
}

//;********************* Write SIO
*****************************************************
int write_w83627UHG_reg(int LDN, int reg, int value)
{
        outportb(INDEX_PORT, 0x07); //LDN register
        delay(1);
        outportb(DATA_PORT, LDN);
        delay(1);
        outportb(INDEX_PORT, reg);
        delay(1);
        outportb(DATA_PORT, value);
        return 0;
}

//;********************* Enter SIO
*****************************************************
int Enter_Configuation()
{
        outportb(INDEX_PORT, 0x87);
        delay(1);
        outportb(INDEX_PORT, 0x87);
        return 0;
}

//;********************* Exit SIO
*****************************************************
int Exit_Configuation()
{
        outportb(INDEX_PORT, 0xaa);                                                     //Exit
W83627THG Configuration
        return 0;
}

//;********************* Init WDTO and SB GPIO
*****************************************************
int Init_WDTO_GPIO_Interface()
{
        Enter_Configuation();
        write_w83627UHG_reg(0x07, 0x30, 0x01);              //;enable GPIO3X
        delay(5);
        write_w83627UHG_reg(0x07, 0xE0, read_w83627UHG_reg(0x07, 0xE0) & 0xF0);
        //Set GPIO30,31,32,33 to output
        delay(5);
        write_w83627UHG_reg(0x07, 0xE0, read_w83627UHG_reg(0x07, 0xE0) | 0xF0);
        //Set GPIO34,35,36,37 to intput
        delay(5);
        Exit_Configuation();

        return 0;
}

//input: none
//output: 1 = not correct SuperIO
```

```
//          0 = correct SuperIO
int detect_sio(void)
{
        //check the SuperIO version
        if(read_w83627UHG_reg(0,CR20_REG)==W83627UHG_ID)
                return 0;
        else
                return 1;
}
```

Terms and Conditions

Date:2001.01.06

Warranty Policy :

1. All products are under warranty against defects in materials and workmanship for a period of one year from the date of purchase.
2. The buyer will bear the return freight charges for goods returned for repair within the warranty period; whereas the manufacturer will bear the after service freight charges for goods returned to the user.
3. The buyer will pay for repair (for replaced components plus service time) and transportation charges (both ways) for items after the expiration of the warranty period.
4. If the RMA Service Request Form does not meet the stated requirement as listed on "RMA Service," RMA goods will be returned at customer's expense.
5. The following conditions are excluded from this warranty:

   Improper or inadequate maintenance by the customer

   Unauthorized modification, misuse, or reversed engineering of the product

   Operation outside of the environmental specifications for the product.

RMA Service :

Requesting a RMA#

1. To obtain a RMA number, log on to the leRMA web interface at http://erma.lannerinc.com . Use you user name and password as supplied to you from your Lanner account manager.

2. Once you have logged on, follow the onscreen instructions to complete the RMA process.

3. Ship the defective unit(s) on freight prepaid terms. Use the original packing materials when possible.

4. Mark the RMA# clearly on the box.

Note: If you have any issues with the eRMA system, email support@lannerinc.com for assistance.

RMA Service Request Form

When requesting RMA service, please fill out the following form. Without this form enclosed, your RMA cannot be processed.

RMA No:	Reasons to Return: □ Repair (Please include failure details) □ Testing Purpose
Company:	Contact Person:
Phone No.	Purchased Date:
Fax No.:	Applied Date:
Return Shipping Address: Shipping by: □ Air Freight □ Sea □ Express ___ □ Others:_____	

Item	Model Name	Serial Number	Configuration

Item	Problem Code	Failure Status

*Problem Code:

01:D.O.A.	07: BIOS Problem	13: SCSI	19: DIO
02: Second Time R.M.A.	08: Keyboard Controller Fail	14: LPT Port	20: Buzzer
03: CMOS Data Lost		15: PS2	21: Shut Down
04: FDC Fail	09: Cache RMA Problem	16: LAN	22: Panel Fail
05: HDC Fail	10: Memory Socket Bad	17: COM Port	23: CRT Fail
06: Bad Slot	11: Hang Up Software	18: Watchdog Timer	24: Others (Pls specify)
	12: Out Look Damage		

Request Party

Confirmed By Supplier

Authorized Signatures / Date

Authorized Signatures / Date